

Amendments to the Specification

Please replace the paragraph that begins on Page 15, line 11 and carries over to Page 16, line 7 with the following marked-up replacement paragraph:

— Fig. 3B illustrates an event annotation of the type disclosed by the present invention. The annotated message 320 has been programmatically created from message 300 by adding additional slots thereto. In this example, the additional slots are shown as elements 330 and 340. In addition, the value of slot 350 is modified. Added slot 330 specifies an executable task that pertains to the event represented by original message 300; added slot 340 indicates that additional tasks or other types of information may also be added, if desired. Slot 330 identifies an inventory registration process that may be invoked, and which (presumably) uses the content of its containing annotated message 320 for input. Update flag [[310]] 350 is provided as a generic means for enabling an EMS to efficiently check its event repository to determine whether events exist which require attention or which can be recycled. This update flag may be provided as part of the transformation process which creates events from messages, as has been discussed, in which case the application generating the message does not need to be aware of the update flag. Note that events may be recycled as many times as desired, even though the update flag may be set to TRUE after a particular processing of that event. (Note that while the examples provide the name of an executable task as the annotation, this is for purposes of illustration and not of limitation. Other types of information may be provided, such as a rule or a simple property that might be associated with an existing rule.) —

Please replace the paragraph on Page 24, lines 2 - 12 with the following marked-up replacement

paragraph:

— One type of annotation that may be performed by Block 685 is a re-annotation of an already-annotated message. This has been previously discussed with reference to programmatically created creating information to invoke unregistration processing for events that were previously annotated to invoke a registration task. As another example, the new task which is being dynamically reflected in the events may be to unregister all Product X, version 1.5 installations from the backup/restore processing system. (Perhaps a site-wide decision has been made that the backup and restore processes should be optimized by omitting backup/restore procedures for this software product.) In this case, the processing of Block 685 preferably comprises changing the value of the slot 330 (see Fig. 3B) or property 410 (see Fig. 4A) to the task which will perform this unregistration, and setting the update flag to FALSE to indicate that this task has not yet been performed. --